**Adding methods/property to existing class using prototype:**

Person. Nationality = “English”;

function Person(first, last, age, eyecolor) {  
    this.firstName = first;  
    this.lastName = last;  
    this.age = age;  
    this.eyeColor = eyecolor;  
}  
Person.prototype.nationality = "English";

**Creating an object:**

methodName : function() {

}

Accessing: objectName.methodName()

**Calling request xmlhttp request:**

var xhttp = new XMLHttpRequest();  
xhttp.onreadystatechange = function() {  
    if (this.readyState == 4 && this.status == 200) {  
       // Typical action to be performed when the document is ready:  
       document.getElementById("demo").innerHTML = xmlhttp.responseText;  
    }  
};  
xhttp.open("GET", "filename", true);  
xhttp.send();

**Creating Arrays and loops:**

var cars = ["Saab", "Volvo", "BMW"];

Loops: for (i = 0; i < cars.length; i++) {   
    text += cars[i] + "<br>";  
}

**Creating functions with arguments:**

Arguments are passed by value.

x = sumAll(1, 123, 500, 115, 44, 88);  
  
function sumAll() {  
    var i, sum = 0;  
    for (i = 0; i < arguments.length; i++) {  
        sum += arguments[i];  
    }  
    return sum;  
}

**Creating methods:**

function Person (Name, age, gender) {  
    this.Name = Name;    
    this.age = age;  
    this.gender = gender;  
    this.changeName = function (name) {  
        this.Name = name;  
    };  
}

**Creating objects using class:**

Class Person {

function Person (Name, age) {

this.Name = Name;

this.age = age;

}

}

var p = new Person(“Bob”, 25);

**Creating properties:**

document.write(p.Name + “has age of” + p.age);

**hosting:**

we can host declarations, not initializations.

Example(this works):

var x = 5;

y = 7;

document.write(“x value is” + x + “and y value is” +y);

var y;

(this doesn’t):

var x = 5;

var y;

document.write(“x value is” + x + “and y value is” +y);

y = 7;

**inheritance:**

var ClassA = function() {

this.name = "class A";

}

ClassA.prototype.print = function() {

console.log(this.name);

}

**Self invoked functions:**

The functions which are invoked before being called.

(function () {

document.getElementById("demo").innerHTML = "Hello! I called myself";

})();

var a = new ClassA();

a.print();

**setIntervel:**

The setInterval() method calls a function or evaluates an expression at specified intervals (in milliseconds).

function myFunction() {

setInterval(function(){ alert("Hello"); }, 3000);

}